An efficient secured CCSDS based Telemetry system for ISRO's near earth and deep space missions

S.Thiruppathirajan[a], D.Sheba Elizabeth[b], C.Preetha, S.Sreekumar, P.Vinod, A.R.Krishnan, T.Mookiah and John P. Zachariah
Vikram Sarabhai Space Centre, ISRO,
Thiruvananthapuram-695022, India
[a]s_thiruppathirajan@vssc.gov.in
[b]d_shebaelizabeth@vssc.gov.in

Abstract— **This paper presents the details of an efficient, adaptive, secured and low complexity featured Consultative Committee for Space Data Systems (CCSDS) based telemetry system developed for ISRO's near earth missions carrying humans onboard and deep space missions to Moon, Mars etc., The present launcher telemetry systems are designed based on custom specific interface standards, protocols and data handling methods which are not suitable for interoperability required for above said missions. Hence there was a strong need to realise a system which will be based on recommendations provided by CCSDS to have end-to-end system solutions for the interoperability and cross-support requirements. CCSDS Telemetry Equipment (CTE) has been developed using in-house developed Digital Signal Processing (DSP) core embedded on an Actel's ProASIC3 Field Programmable Gate Array (FPGA) to cater to the major services of telemetry system such as source coding, packet telemetry, channel coding and encryption.**

Keywords — CCSDS; lossless data compression; packets; advanced encryption standard; Turbo coding.

I. Introduction

The onboard telemetry systems of ISRO's launchers are presently based on Time Division Multiplexed Pulse Code Modulation system. This system acquires data and telemeters in fixed sampling rates using custom devised interface standards, protocols and data handling methods. Such a custom designed telemetry system can impose constraints for interoperable missions such as Human in space missions, docking missions and deep space missions. Hence it was required to develop a telemetry system using resources recommended by Consultative Committee for Space Data Systems (CCSDS) which enhances mission interoperability and cross-support.

CCSDS provides well-engineered and sound technical solutions that enhance interoperability and cross-support among the space faring participants, while also reducing risk, project cost, and development time.

The major features of the developed CCSDS based telemetry system are adaptive source coding for lossless data compression, packet telemetry to format the variable length data stream into a fixed-length frame easing transportation from space to ground, data encryption providing security measures to have authenticated transportation of telemetry data and channel coding to reduce the bit error rate considerably during telemetry data communication to the ground.

II. Need for CCSDS recommended telemetry services

With the fixed available telemetry bandwidth, it is difficult to cater to the increased number of instruments for measurements, and the near static nature of instrument data occupies major portion of available bandwidth. Hence, compression is required to reduce the large data rate consumed by several static data and to have more information with the available bandwidth. Information loss due to compression is unacceptable especially for missions carrying humans onboard and deep space missions. This calls for lossless data compression to meet the telemetry requirements. CCSDS recommends extended RICE algorithm [1] as the de facto standard for lossless data compression.

The essential purpose of the packet telemetry system [2] is to permit the various application processes onboard to create units of data called source packets. These packets are then transmitted to ground via a communication channel in a way that enables the ground system to recover the data with high reliability. Packet telemetry also provides a mechanism for implementing common data structures and protocols which can enhance the development and operation of mission services.

Channel coding enables processing data in such a way that distinct messages are created which are easily distinguishable from one another for reliable transportation of data to ground in the presence of channel induced noise. This allows reconstruction of the data on ground with low bit error probability. CCSDS recommends Turbo encoding [4] for efficient and reliable communication on a power limited channel.

For missions carrying human onboard there are concerns regarding the confidentiality of information conveyed onboard from ground and vice-versa across the unsecured channel. Encryption on telemetry data protects its confidentiality and provides data integrity over unsecured channel. Of the various encryption options available, CCSDS recommends the 128 bit Advanced Encryption Standard Rijndael algorithm [3] as the suitable candidate for encryption.

III. CCSDS Telemetry System Concept

CCSDS Telemetry System is broken down into two concepts namely "Packet Telemetry [2]" and "Telemetry

Channel Coding [4]". The CCSDS telemetry system concept is based on a technique known as “layering” which is a very useful tool for transforming the telemetry system concept into sets of operational and formatting procedures. Fig. 1 illustrates the CCSDS Telemetry System in terms of a layered service model.

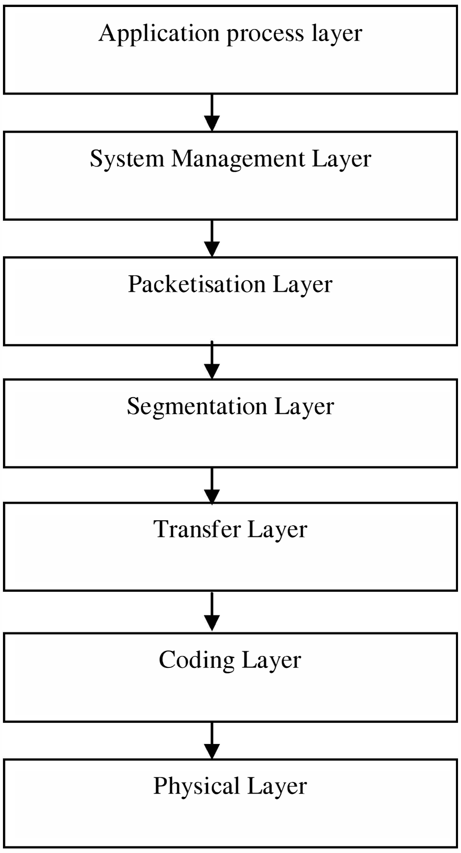

Figure. 1 CCSDS layered service model

The application process layer and system management layer together generate a method to investigate required telemetry parameters using instruments onboard for data collection and analysis and to translate the physical measurements into sets of application data units. The packetisation layer provides end-to-end delivery of application data units using packets. The segmentation layer is an optional layer for segmenting and transferring longer packets through a space data channel. Transfer layer provides reliable transferring of packets in a common structure called transfer frame. Coding layer protects the transfer frames against errors induced during transmission through the noisy physical communications channel and offers security to the transfer frame transmitted. Physical layer provides the physical connection i.e. Radio Frequency (RF) communication between onboard telemetry system and the ground station.

The CCSDS based telemetry system as explained in the following sections contains various Data Acquisition Engines (DAEs) housed with application process and system management layers and CCSDS Telemtery Equipment (CTE) housed with packetisation layer for performing source coding and packetisation, transfer layer and coding layer to protect the telemetry data.

IV. Design of CCSDS based Telemetry System

The onboard hardware of the CCSDS based telemetry system developed for ISRO’s missions as shown in Fig. 2 consists of several DAEs running different application processes for acquiring front-end instruments originating data. DAEs feature software programmable amplification, offset, filtering and sensor excitation to acquire and measure physically variable measurements like force, pressure, strain, temperature, voltage, current, etc., These DAEs are controlled by commands from CTE. CTE collects the data from all the DAEs and processes the data as per CCSDS recommendation for compression, coding and encryption and sends it to ground station in the CCSDS recommended packet format through a RF transmitter.

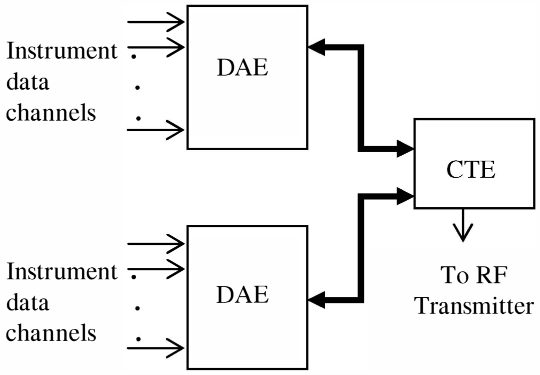

Figure. 2 Onboard CCSDS based Telemetry System

CTE is designed with an in-house developed Digital Signal Processing (DSP) core on an Actel’s ProASIC 3 Field Programmable Gate Array (FPGA) housing necessary interfaces with DAEs as shown in Fig. 3 and logic functions for above mentioned CCSDS services.

Each DAE generates data at different rates corresponding to various application processes, which can be grouped to a programmable number of maximum 8 nos. of virtual channels. Virtual channel [2] is a concept of having sequences of telemetry transfer frame originating from same or multiple DAEs. Time division multiplexing is done to give access for each virtual channel to the physical channel.

CTE communicates with the DAEs and onboard computer to collect the data through custom made interfaces of maximum 1Mbps speed and standard interfaces like Mil-Std 1553B/RS485 of maximum 2Mbps speed.

The acquired data are stored in internal buffers of CTE. In-house developed 16-bit fixed point processor core resides in the FPGA along with other logic and shares the resources like EEPROM and RAM. The embedded software is used in the implementation of CCSDS recommendations, viz, compression, packetization, Turbo coding and encryption. Data compression is done based on extended RICE algorithm on selected data. After processing of the DAE data, the data generated from various application processes are packetized into various source packets by the packet generator. The output of the packet generator is sent to the Virtual Channel Transfer Frame (VCTF) generator which generates the VCTFs based on the availability of data from different virtual channels. The Master Channel Transfer Frame (MCTF) generator generates MCTF by selecting the VCTFs based on the priority assigned for each virtual channel. MCTF generator writes MCTF as uncoded transfer frames of 223 octets and is designed such that the output data rate is maintained constant for a particular mission phase.

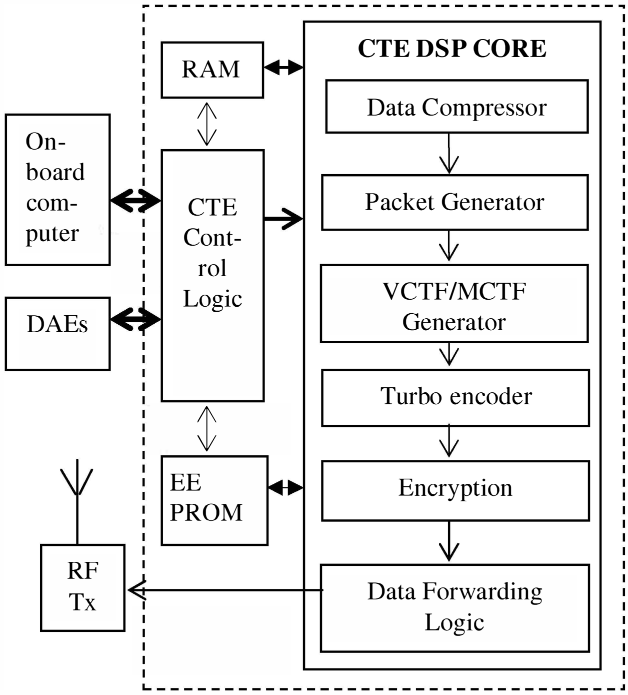

Figure. 3 CTE Block diagram

Each uncoded transfer frame is Turbo coded and an Attached Sync Marker (ASM) of 64-bits is attached. As conventional software approach will not be able to meet the coding/interleaving requirements at higher bit rates, dedicated hardware acceleration is provided to the DSP architecture in the form of Algorithmic instruction supporting simultaneous interleaving and coding. This is an iterative instruction generating the turbo codes for each 8 bits data. The processes involved for each bit (per iteration) are the generation of the coded outputs for both component encoders and computation of interleaved address for the next bit.

Embedded software for 128 Bit AES encryption is executed over the telemetry transfer frame. The encrypted data is given to the data forwarding logic, which generates serial stream of data at the specified data rate. This will be fed to the transmitter for modulation.

This design targeted to FPGA has a utilization of about 700K gates. This implementation is capable of data rates maximum up to 2Mbps.

A. *Implementation of CCSDS recommended lossless data compression in CTE*

The extended RICE algorithm is recommended as the de facto lossless data compression algorithm by CCSDS, because of its high speed for real time data processing, quick adaptation to varying statistics and low implementation complexity [1].

Data compression is achieved by applying the extended RICE algorithm to the acquired data from DAEs and CTE's own data stored in the buffers. Using dual buffering scheme, acquired data is stored in one of the buffers and from the other buffer stored data is compressed. At a time, a block of stored data of a particular instrument data channel of an application process is considered. The block size is programmable and varies from 16 data samples to 128 data samples depending on the sampling rate of a particular channel.

Extended RICE algorithm is implemented onboard with two separate functional blocks namely preprocessor and adaptive entropy coder as shown in Fig.4. Preprocessor contains a very simple linear one-dimensional predictor, which simply predicts that the next sample will be the same as the last, followed by prediction error mapper. The prediction error results in removal of redundancy between the adjacent samples. These errors are statistically independent to each other so that better coding efficiency can be achieved. Mapper maps these errors to 8 bits positive integers on which adaptive entropy encoder works.

DSP core embedded in the FPGA executes the mapped value from the instrument data using (1) and (2) as follows. Let x be the current sample of the instrument data and x^ be the predicted one for x and usually be the previous sample. Let Δ be the one-dimensional predictor output.

$$\Delta = x\text{-}x\hat{} \qquad (1)$$

The mapped value 'm' from Δ is

$$\textit{if}\,(\Delta \geq 0)\ \{\ \textit{if}\,(\Delta \leq x\hat{})\ m=2\Delta;\ \textit{else}\ m=x;\}$$

$$\textit{else}\ \{\ \textit{if}\,(|\Delta|\leq 255\text{-}x\hat{})\ m=2|\Delta|\text{-}1;\ \textit{else}\ m=255\text{-}x\} \qquad (2)$$

Adaptive encoder contains many variable length encoders where each one codes with better efficiency depending on the source entropy. The different variable length encoders are namely "option zero block", "option second extension", "option Fundamental Sequence (FS)", "option split-sample" and "option no compression". Option zero block and second extension option are suitable for coding low entropy up to 1.5 bits per sample (highly compressible) instrument data, where as option FS and split sample options are suitable for moderate entropy (less compressible) source data and no compression option for coding high entropy (random data and no compression) source data. For the mapped value 'm', option FS encodes to 'm' zeroes followed by a delimiter '1'. The k^{th} split-sample option takes a block of mapped value, splits off the *k* least significant bits from each mapped value and encodes the remaining higher order bits with a simple FS codeword before appending the split bits to the encoded FS codeword. Option zero block is exercised when all the mapped values of a block are zeroes. The extended RICE algorithm tries to get most efficient encoding option out of the above mentioned options in terms of Compression Ratio (CR). The search for the best encoding option is performed before the encoding stage and encoding is performed using that option which is alternate to the CCSDS recommendation but the processing time is significantly reduced. The compressed data as a result of encoding is encapsulated into CCSDS telemetry packets along with the application process identifier (APID) and the selected compression option.

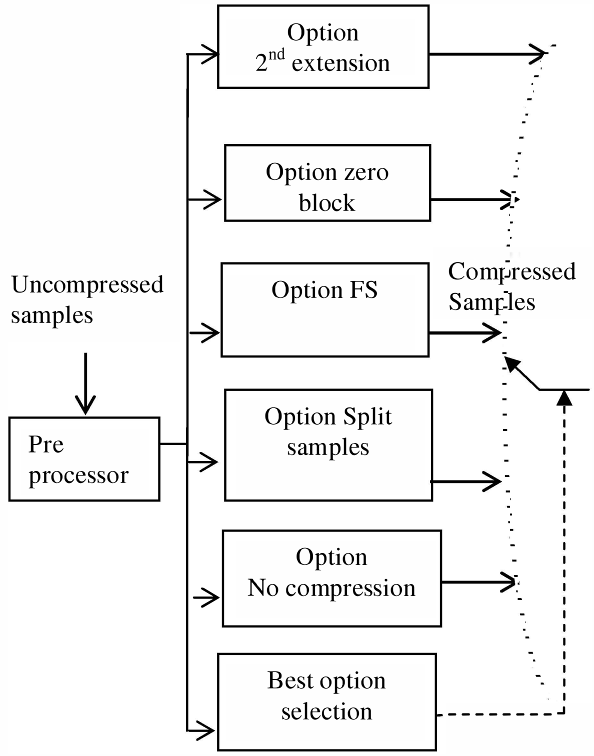

Figure. 4 Extended RICE Algorithm logic block

B. *Implementation of CCSDS recommended packetisation in CTE*

The packet telemetry concept permits onboard sources to transmit these data to ground over a communication channel in a way that permits the ground system to recover each data unit with high reliability. The data from onboard sources may include data from onboard computers, compressed data, and data from other sub-systems. To accomplish the functions of packet telemetry, two data structures, viz., source packets and transfer frames, and a multiplexing process to interleave source packets from various application processes into transfer frames are used. The typical telemetry data flow of the CCSDS based telemetry system is shown in Fig. 5.

A "source packet" [2] is the basic data unit telemeters to the ground and generally contains a meaningful quantity of related measurements from a particular source. Each source packet consists of header and data field. Source packet header identifies the source and characteristics of the packet. The internal data content of the source packet is completely under the control of the application process. Source packets are generated at fixed or variable intervals and may be fixed or variable in length. Hence each application process optimises the size and structure of its data unit based on the requirement.

The transfer frame [2] serves as an envelope for transmitting the source packets over a noisy space-to-ground channel. Each transfer frame consists of header and data field. The header carries information that permits the ground system to route the transfer frames to their intended destination. The transfer frame is of fixed length.

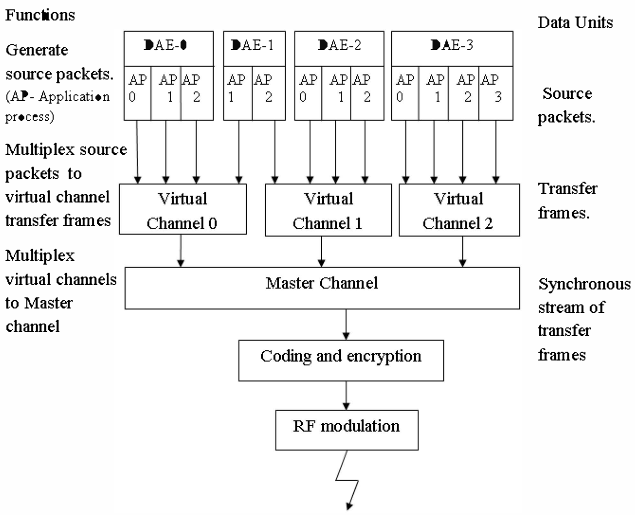

Figure. 5 CCSDS based onboard telemetry flow

Multiple, individual, asynchronous application processes running onboard a space vehicle generates data at different rates. These data are received by CTE as and when they come. CTE also has application processes running internally, generating data like compressed data. The data from various sources are converted into source packets in CTE.

The header includes the APID to know about the origin of the data. The continuity of the data can be ensured using source sequence count which also forms part of the header. Since there are multiple users and the space communication system is capacity-limited, the users are given access to the channel in a time shared manner. This is achieved by a process called virtual channelization. In this scheme the application processes are grouped into different virtual channels, based on the characteristics of the data, bit rate, etc. There can be a maximum of eight virtual channels. Depending on the type of data each virtual channel is assigned with a transmission capacity, on a frame by frame basis, to access the physical channel.

The source packets are then multiplexed in CTE to generate virtual channel transfer frames. Short packets may be contained in a single frame, while longer ones may span two or more frames. Since a packet can begin or end at any place in a frame, the entire data field of every frame can be used to carry data; there is no need to tune the sizes of packets or their order of occurrence to fit the frames. When a frame is released, if insufficient source packets are available, idle packet will be sent so as to meet the transfer frame size.

These virtual channel transfer frames are then multiplexed into a synchronous stream of fixed length coded transfer frames for reliable transmission through the physical channel to the ground as shown in Fig. 6. Each transfer frame is identified as belonging to one of the eight virtual channels.

The header will contain information about the virtual channel, so that the virtual channels are demultiplexed in ground and source packets are extracted. The source packets extracted can be distributed to the respective sink processes based on the APID found as part of source packet header.

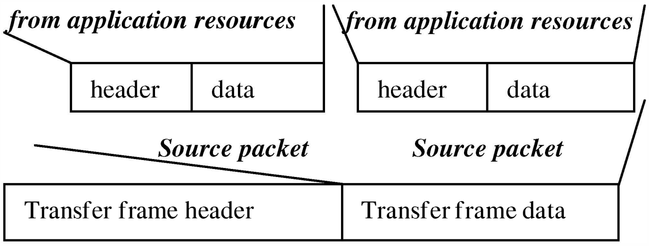

Figure. 6 Packet data structure

If no data is available at the point of release of a transfer frame, idle data will be sent in the transfer frame. This is implemented to keep the data capture element in synchronization in the absence of data. On the ground, the information in the frame and packet headers allows the data acquisition system to extract packets in a standardised way. The transfer frames are stored in buffers available in the hardware memory resources. The transfer frame size is chosen so as to be compatible with CCSDS recommended Turbo coding.

C. Implementation of CCSDS recommended Turbo coding in CTE

When one complete transfer frame has been generated, CCSDS recommended rate ½ turbo coding is done as shown in Fig. 7. Turbo codes achieve near Shannon-limit error correction performance with parallel concatenation of two simple recursive constituent convolutional codes separated by an interleaver. It was reported [5] that Energy per bit to Noise power spectral density (E_b/N_o) of 0.7 dB was required for Bit Error Rate (BER) of 10^{-5} and code rate of ½ which is only 0.7 dB away from the Shannon-limit for the same code rate. CCSDS recommended turbo code accepts a block of *k*, a multiple of 1784 (223 bytes) information bits. Each constituent code generates a set of parity bits. The interleaver permutes the original k information bits before encoding the second code. The interleaver is well-chosen so that information blocks that correspond to error-prone code words in one code will correspond to error-resistant code words in the other code.

Here recursive systematic convolutional encoders are realised using feedback shift registers. For the rate ½ code, the output sequence is "(out1, out2, out1, out3)", repeated *(1784+4)/2* times i.e., puncturing is done for rate ½ coding. Both encoders run for *1784+4* bit times for each code blocks producing *(1784+4)/2* output symbols. For the last 4 bit times, 0s are shifted in and thereby the shift registers are initialised to 0s for the next block (trellis termination). The key feature of turbo encoder is an interleaver (π), which permutes (scrambles) bit-wise the original *k* information bits before input to the second encoder.

In-house developed DSP core performs turbo coding on each transfer frame. The transfer frame length is programmable and any transfer frame length recommended by CCSDS can be used. An application specific instruction is used which efficiently computes the permutation bit address based on the algorithm recommended by CCSDS and generates the turbo code. Due to this approach Turbo coding is implemented using simple software with execution efficiency comparable to the hardware approach. This provides greater advantage for coding at higher bit rates. The coded transfer frame is written in buffer.

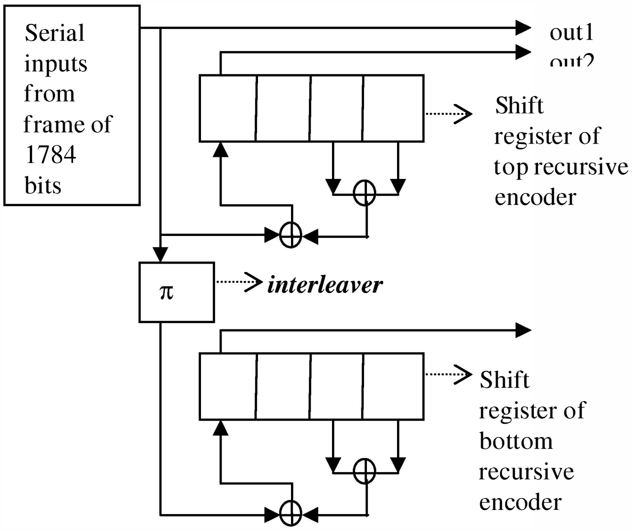
Serial inputs from frame of 1784 bits
out1
out2
Shift register of top recursive encoder
π
interleaver
Shift register of bottom recursive encoder

Figure. 7 Simplified structure of rate ½ Turbo encoder

Proper decoding of the turbo coded frame requires code block synchronization. This is achieved by the usage of Attached Sync Marker (ASM). As per the CCSDS recommendations, ASM should contain *32/r* bits, where *r* is the code rate. Since the turbo encoder implemented uses rate ½ coding, 64 bits of ASM as mentioned in the standard are added prior to each coded transfer frame.

D. Implementation of CCSDS recommended AES algorithm for data encryption in CTE

Advanced Encryption Algorithm (AES) [6] developed by Rijndael is the recommended standard based on its performance on hardware and software and less memory and computation resources requirements as it works on logical XOR and shifting operations. AES is a symmetric, block-cipher algorithm operating over a 128 bits block. The algorithm assumes 128 bits plain data (data to be encrypted) which results in the output of 128 bits of cipher data (encrypted data). Implementation of security services above the telemetry transfer frame is considered here.

Using counter mode technique the transfer frame data to be encrypted is not run through the AES algorithm. Rather, a counter which has been combined with a cipher key is used as the starting input to the algorithm, which in turn produces 128 bit random key blocks. The output bits are XORed with the plain text data to produce the output cipher blocks as shown in Fig. 8. The counter value differs for each 128 bits of transfer frame.

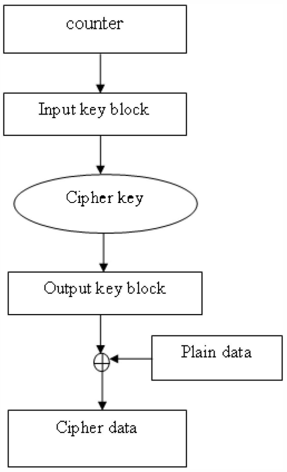

Figure. 8 AES Logic flow

V. Test Results

The performance of extended RICE Algorithm was analysed using flight data of several missions. Instruments generated data acquired was compressed in each Super Frame (SF) which is a time interval of 500 ms. Average Compression Ratio (ACR) of a few instrument measurements for full flight duration is tabulated below.

TABLE I. FEW MEASUREMENTS AND THEIR ACR

Measurement	Sampling rate (per second)	ACR
Engine gimbal drive voltage	500	5.0
Gas motor Pressure	250	3.1
Actuator pressure	125	6.4
Temperature	125	5.3
Axial mode acceleration	500	3.3
Current monitoring	1560	3.9

On an average, 3:1 CR could be achieved by lossless data compression. Variation of CR in every SF throughout the flight duration for Engine gimbal drive voltage is shown in Fig. 9. It can be observed that during quiescent period a maximum CR of about 16:1 is obtained. During staging events, a minimum CR of 1:1 is obtained because of “no compression option” selection.

To evaluate the performance of turbo coding, 1000 nos. of telemetry transfer frame of 1784 bits (223 bytes) were coded and encoded data was Phase Shift Keying (PSK) modulated and transmitted over an Additive White Gaussian Noise (AWGN) channel. The transmitted data was decoded using Maximum A Posteriori (MAP) decoding and BER was computed. The results were compared with the detection of uncoded data. From the BER performance, it is clear that coding gain margin for Turbo codes improves over uncoded one as the no. of iteration for the turbo decoding increases. E_b/N_o of 1.0 dB and a coding gain of 8 to 9 dB for the BER of 10^{-5} was achieved with 9 nos. of iterative decoding.

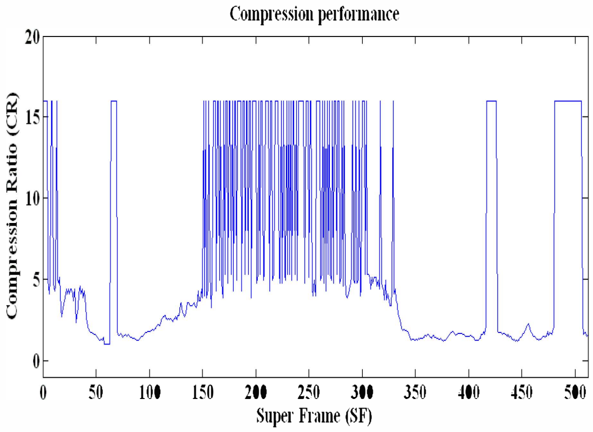

Figure. 9 Engine gimbal drive voltage (SF vs CR)

VI. Conclusion

The new CCSDS based telemetry system facilitates transmission of space acquired data from source to user in the ground in a standardized and highly automated manner. The lossless data compression using extended RICE algorithm results in an average compression ratio of 3:1 and the Turbo channel coding attains 8 to 9 dB coding gain. Thus by use of an efficient DSP embedded processor core on FPGA based hardware, the demands for greater Telemetry throughput for the near earth and deep space missions have been met using CCSDS recommended RICE algorithm, packet telemetry and Turbo coding. Requirement of providing security measures using AES encryption standard also has been adequately addressed.

References

[1] “Lossless Data Compression”, Report Concerning Space Data System Standards, CCSDS 120.0-G-2. Green Book. Issue 1. Washington, D.C.: CCSDS, December 2006.

[2] “Packet Telemetry”, Recommendation for Space Data System Standards, CCSDS 102.0-B-5, Blue Book, Issue 5, Washington, D.C.: CCSDS, November 2000.

[3] “Symmetric Encryption”, Draft Recommendation for Space Data System Practices CCSDS 353.0-R-1. Red Book, Issue 1. Washington, D.C.: CCSDS, October 2008.

[4] “TM Synchronization and Channel Coding”, Recommendation for Space Data System Standards, CCSDS 131.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon Limit Error Correcting Coding: Turbo Codes.” *Proc. 1993 IEEE International Conference on Communications, pp. 1064–1070.*

[6] Advanced Encryption Standard (AES). *Federal Information Processing Standards Special Publication 197. Gaithersburg, Maryland: NIST, 2001. <http://csrc.nist.gov/publications/fips/fips197/fips-97.pdf>*